

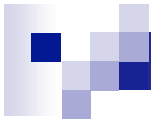


# Artificial Neural Network for Speech Recognition

Austin Marshall

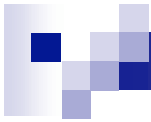
March 3, 2005

2nd Annual Student Research Showcase



# Overview

- ✓ Presenting an Artificial Neural Network to recognize and classify speech
  - ◆ Spoken digits
    - ✓ “one”, “two”, “three”, etc...
- ✓ Choosing a speech representation scheme
- ✓ Training Perceptron
- ✓ Results



# Representing Speech

## ✓ Problem

- ◆ Recording samples never produce identical waveforms
  - ✓ Length
  - ✓ Amplitude
  - ✓ Background noise
  - ✓ Sample rate
- ◆ However, perceptual information relative to speech remains consistent

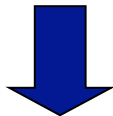
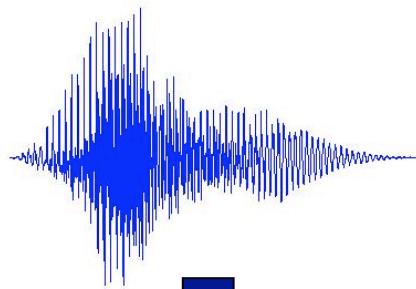
## ✓ Solution

- ◆ Extract speech-related information
  - ✓ See: Spectrogram

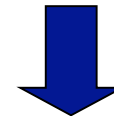
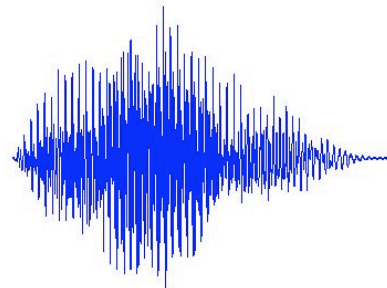
# Representing Speech

Waveform

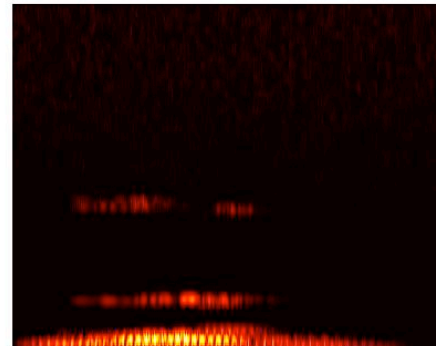
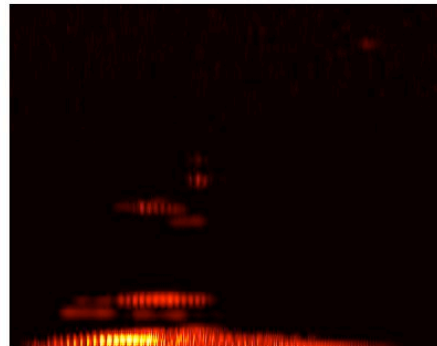
“one”



“one”

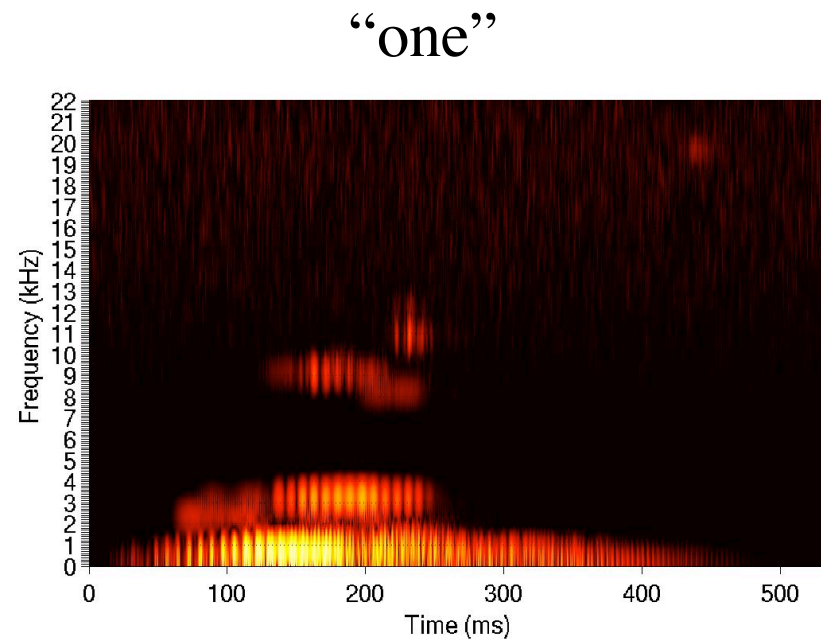


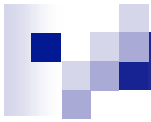
Spectrogram



# Spectrogram

- ✓ Shows change in amplitude spectra over time
- ✓ Three dimensions
  - ◆ X Axis: Time
  - ◆ Y Axis: Frequency
  - ◆ Z axis: Color intensity represents magnitude



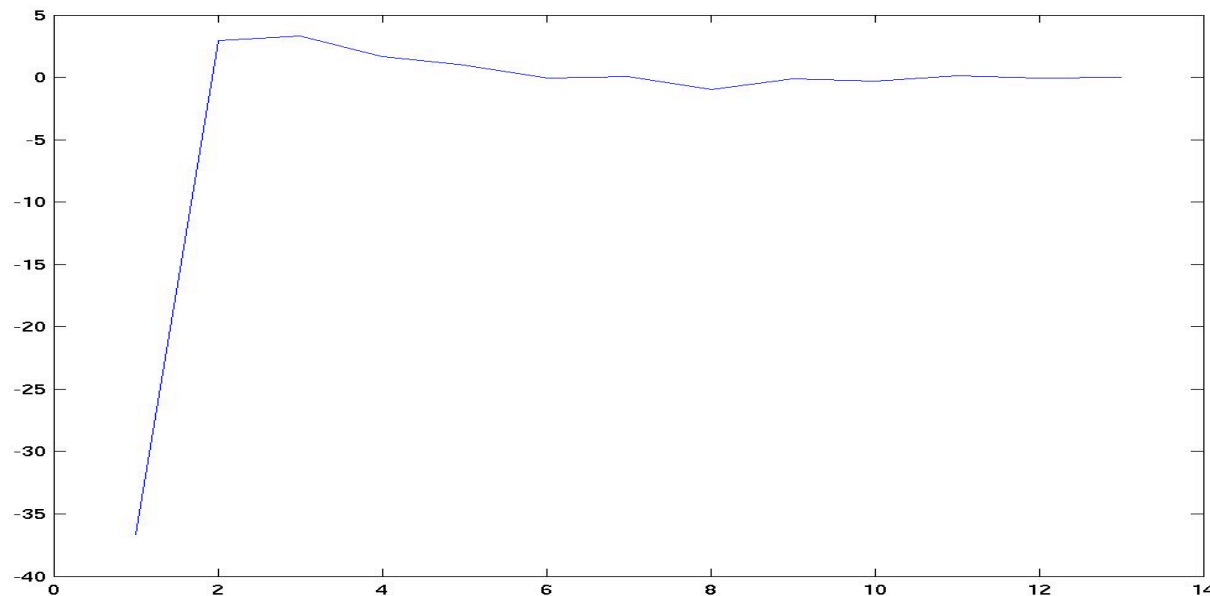


# Mel Frequency Cepstrum Coefficients

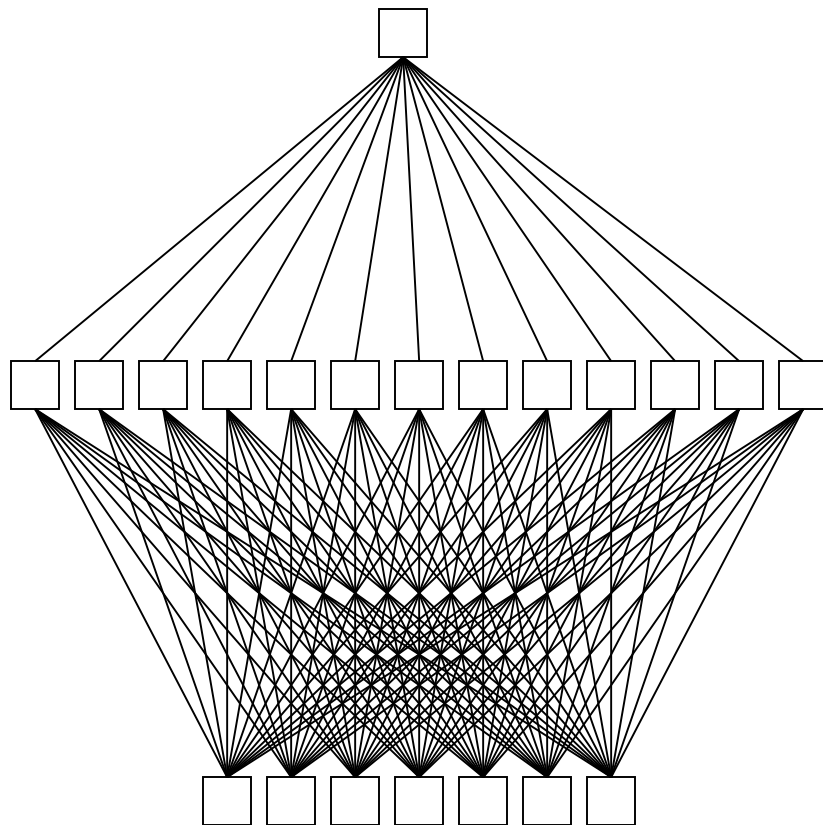
- ✓ Spectrogram provides a good visual representation of speech but still varies significantly between samples
- ✓ A cepstral analysis is a popular method for feature extraction in speech recognition applications, and can be accomplished using Mel Frequency Cepstrum Coefficient analysis (MFCC)

# Mel Frequency Cepstrum Coefficients

- ✓ Inverse Fourier transform of the log of the Fourier transform of a signal using the Mel Scale filterbank
- ✓ mfcc function returns vectors of 13 dimensions



# Network Architecture

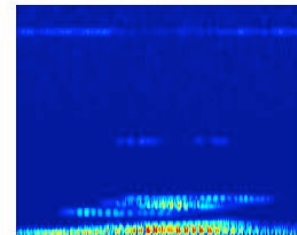
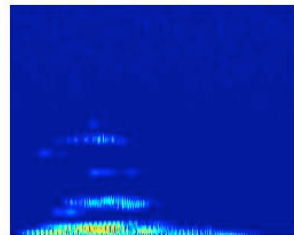
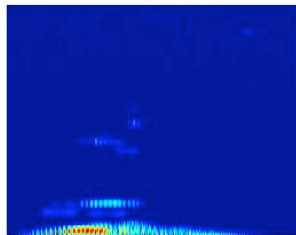


- ✓ Input layer
  - ◆ 26 Cepstral Coefficients
- ✓ Hidden Layer
  - ◆ 100 fully-connected hidden-layer units
  - ◆ Weight range between -1 +1
    - ✓ Initially random
    - ✓ Remain constant
- ✓ Output
  - ◆ 1 output unit for each target
  - ◆ Limited to values between 0 and +1

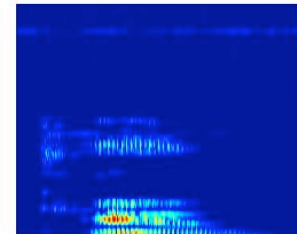
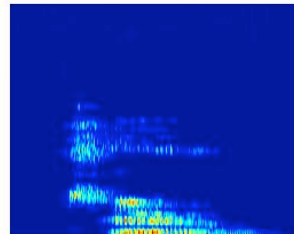
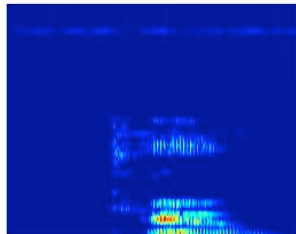


# Sample Training Stimuli (Spectrograms)

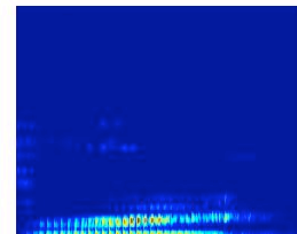
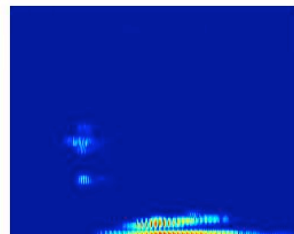
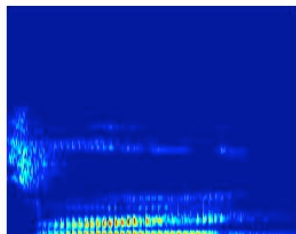
“one”

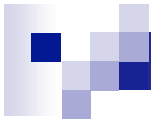


“two”



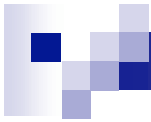
“three”





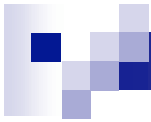
# Training the network

- ✓ Spoken digits were recorded
  - ◆ Seven samples of each digit
  - ◆ “One” through “eight” recorded
  - ◆ Total of 56 different recordings with varying lengths and environmental conditions
- ✓ Background noise was removed from each sample



# Training the network

- ✓ Calculate MFCC using Malcolm Slaney's Auditory Toolbox
  - ◆ `c=mfcc(s,fs,fix((3*fs)/(length(s)-256)))`
  - ◆ Limits frame rate such that mfcc always produces a matrix of two vectors corresponding to the coefficients of the two halves of the sample
- ✓ Convert 13x2 matrix to 26 dimensional column vector
  - ◆ `c=c(:)`

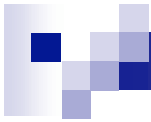


# Training the network

- ✓ Supervised learning

- ◆ Choose intended target and create a target vector
- ◆ 56 dimensional target vector

- ✓ If training the network to recognize spoken “one”, target has a value of +1 for each of the known “one” stimuli and 0 for everything else



# Training the network

- ✓ Train a multilayer perceptron with feature vectors (simplified)
  - ◆ Select stimuli at random
  - ◆ Calculate response to stimuli
  - ◆ Calculate error
  - ◆ Update weights
  - ◆ Repeat
- ✓ In a finite amount of time, the perceptron will successfully learn to distinguish between stimuli of an intended target and not.

# Training the network

- ✓ Calculate response to stimuli

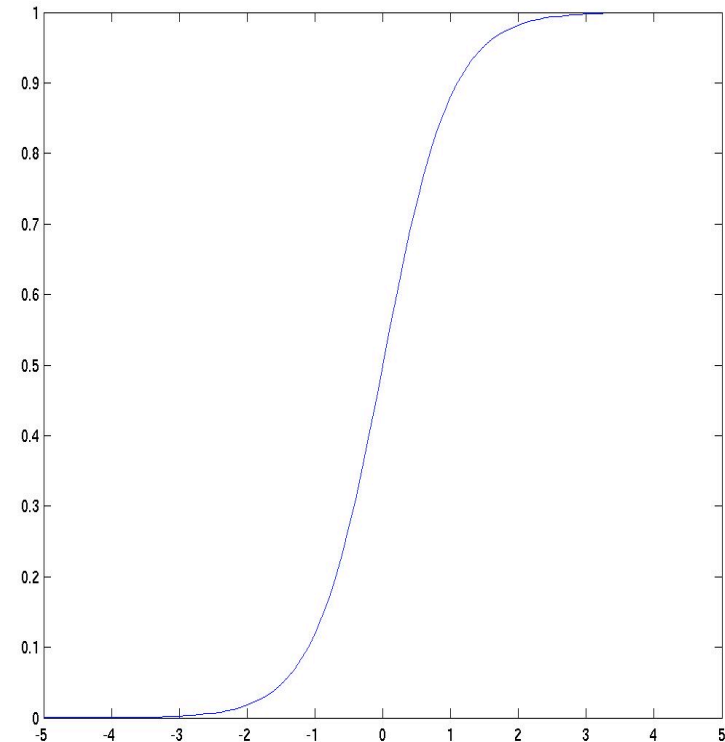
- ◆ Calculate hidden layer
    - ✓  $\mathbf{h} = \text{sigmoid}(\mathbf{W}^* \mathbf{s} + \text{bias})$

- ◆ Calculate response
    - ✓  $o = \text{sigmoid}(\mathbf{v}^* \mathbf{h} + \text{bias})$

- ✓ Sigmoid transfer function

- ◆ Maps values between 0 and +1

$$\text{sigmoid}(x) = 1 / (1 + e^{-x})$$

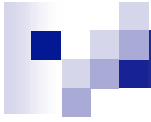




# Training the network

## ✓ Calculate error

- ◆ For a given stimuli, error is the difference between target and response
- ◆  $t-o$
- ◆  $t$  will be either 0 or 1
- ◆  $o$  will be between 0 and +1



# Training the network

## ✓ Update weights

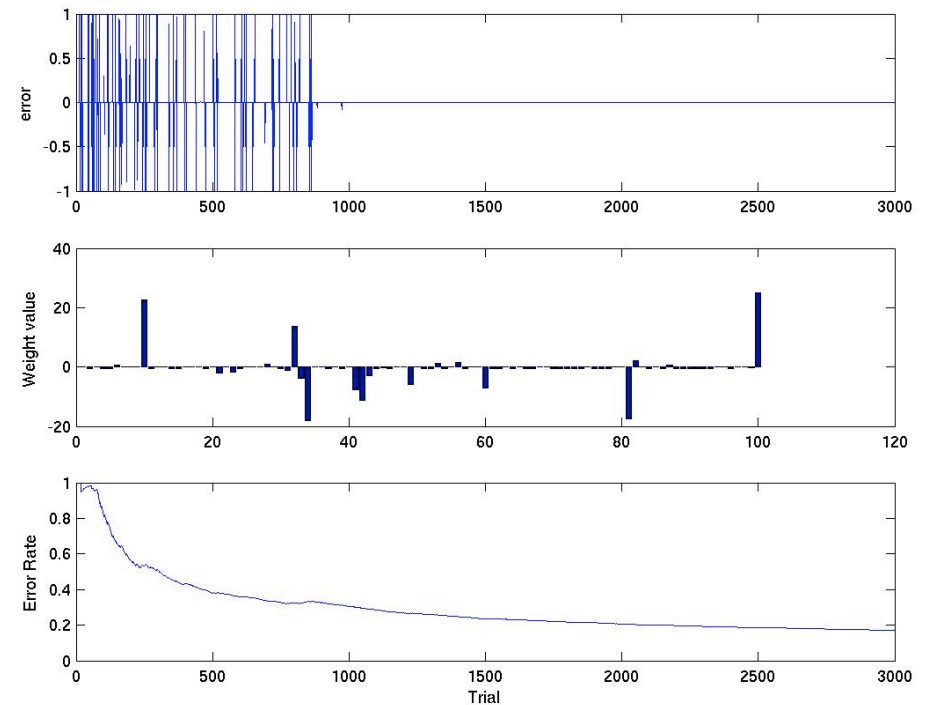
- ◆  $\mathbf{v} = \mathbf{v}_{\text{previous}} + \gamma(t - o)\mathbf{h}^T$
- ◆  $\mathbf{v}$  is weight vector between hidden-layer units and output
- ◆  $\gamma$  (gamma) is learning rate



# Results

- ✓ Learning rate: +1
- ✓ Bias: -1
- ✓ 100 hidden-layer units
- ✓ 3000 iterations
- ✓ 316 seconds to learn target

Target = “one”

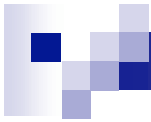




# Results

## ✓ Response to unseen stimuli

- ◆ Stimuli produced by same voice used to train network with noise removed
- ◆ Network was tested against eight unseen stimuli corresponding to eight spoken digits
- ◆ Returned 1 (full activation) for “one” and zero for all other stimuli.
- ◆ Results were consistent across targets
  - ✓ i.e. when trained to recognize “two”, “three”, etc...
- ◆  $\text{sigmoid}(v * \text{sigmoid}(w * t1 + \text{bias}) + \text{bias}) == 1$



# Results

- ✓ Response to noisy sample
  - ◆ Network returned a low, but response  $> 0$  to a sample without noise removed
- ✓ Response to foreign speaker
  - ◆ Network responded with mixed results when presented samples from speakers different from training stimuli
- ✓ In all cases, error rate decreased and accuracy improved with more learning iterations



# References

- ✓ Jurafsky, Daniel and Martin, James H. (2000) *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition* (1st ed.). Prentice Hall
- ✓ Golden, Richard M. (1996) *Mathematical Methods for Neural Network Analysis and Design* (1st ed.). MIT Press
- ✓ Anderson, James A. (1995) *An Introduction to Neural Networks* (1st ed.). MIT Press
- ✓ Hosom, John-Paul, Cole, Ron, Fanty, Mark, Schalkwyk, Joham, Yan, Yonghong, Wei, Wei (1999, February 2). *Training Neural Networks for Speech Recognition* Center for Spoken Language Understanding, Oregon Graduate Institute of Science and Technology, [http://speech.bme.ogi.edu/tutordemos/nnet\\_training/tutorial.html](http://speech.bme.ogi.edu/tutordemos/nnet_training/tutorial.html)
- ✓ Slaney, Malcolm *Auditory Toolbox* Interval Research Corporation